

# ROM 的应用

## 1. 作函数运算表电路

数学运算是数控装置和数字系统中需要经常进行的操作，如果事先把要用到的基本函数变量在一定范围内的取值和相应的函数取值列成表格，写入只读存储器中，则在需要时只要给出规定“地址”就可以快速地得到相应的函数值。这种 ROM，实际上已经成为函数运算表电路。

【例 1】试用 ROM 构成能实现函数  $y=x^2$  的运算表电路， $x$  的取值范围为 0~15 的正整数。

【解】

① 设定变量

自变量  $x$  的取值范围为 0~15 的正整数，对应的 4 位二进制正整数，用  $B=B_3B_2B_1B_0$  表示。根据  $y=x^2$  的运算关系，可求出  $y$  的最大值是  $15^2=225$ ，可以用 8 位二进制数  $Y=Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$  表示。

② 列真值表—函数运算表

表 1 例 1 中 Y 的真值表

| $B_3$ | $B_2$ | $B_1$ | $B_0$ | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ | 十进制数 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0    |
| 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 1    |
| 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 4    |
| 0     | 0     | 1     | 1     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 1     | 9    |
| 0     | 1     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 16   |
| 0     | 1     | 0     | 1     | 0     | 0     | 0     | 1     | 1     | 0     | 0     | 1     | 25   |
| 0     | 1     | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 1     | 0     | 0     | 36   |
| 0     | 1     | 1     | 1     | 0     | 0     | 1     | 1     | 0     | 0     | 0     | 1     | 49   |
| 1     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 64   |
| 1     | 0     | 0     | 1     | 0     | 1     | 0     | 1     | 0     | 0     | 0     | 1     | 81   |
| 1     | 0     | 1     | 0     | 0     | 1     | 1     | 0     | 0     | 1     | 0     | 0     | 100  |
| 1     | 0     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 0     | 0     | 1     | 121  |
| 1     | 1     | 0     | 0     | 1     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 144  |
| 1     | 1     | 0     | 1     | 1     | 0     | 1     | 0     | 1     | 0     | 0     | 1     | 169  |
| 1     | 1     | 1     | 0     | 1     | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 196  |
| 1     | 1     | 1     | 1     | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 1     | 225  |

③ 写标准与或表达式

$$Y_7 = m_{12} + m_{13} + m_{14} + m_{15}$$

$$Y_6 = m_8 + m_9 + m_{10} + m_{11} + m_{14} + m_{15}$$

$$Y_5 = m_6 + m_7 + m_{10} + m_{11} + m_{13} + m_{15}$$

$$Y_4 = m_4 + m_5 + m_7 + m_9 + m_{11} + m_{12}$$

$$Y_3 = m_3 + m_5 + m_{11} + m_{13}$$

$$Y_2 = m_2 + m_6 + m_{10} + m_{14}$$

$$Y_1 = 0$$

$$Y_0 = m_1 + m_3 + m_5 + m_7 + m_9 + m_{11} + m_{13} + m_{15}$$

#### ④画 ROM 存储矩阵节点连接图

为做图方便，可将 ROM 矩阵中的二极管用节点表示。

在图 1 所示电路中，字线  $W_0 \sim W_{15}$  分别与最小项  $m_0 \sim m_{15}$  一一对应，我们注意到作为地址译码器的与门阵列，其连接是固定的，它的任务是完成对输入地址码（变量）的译码工作，产生一个个具体的地址—地址码（变量）的全部最小项；而作为存储矩阵的或门阵列是可编程的，各个交叉点—可编程点的状态，也就是存储矩阵中的内容，可由用户编程决定。

当我们把 ROM 存储矩阵做一个逻辑部件应用时，可将其用方框图表示如图 2。

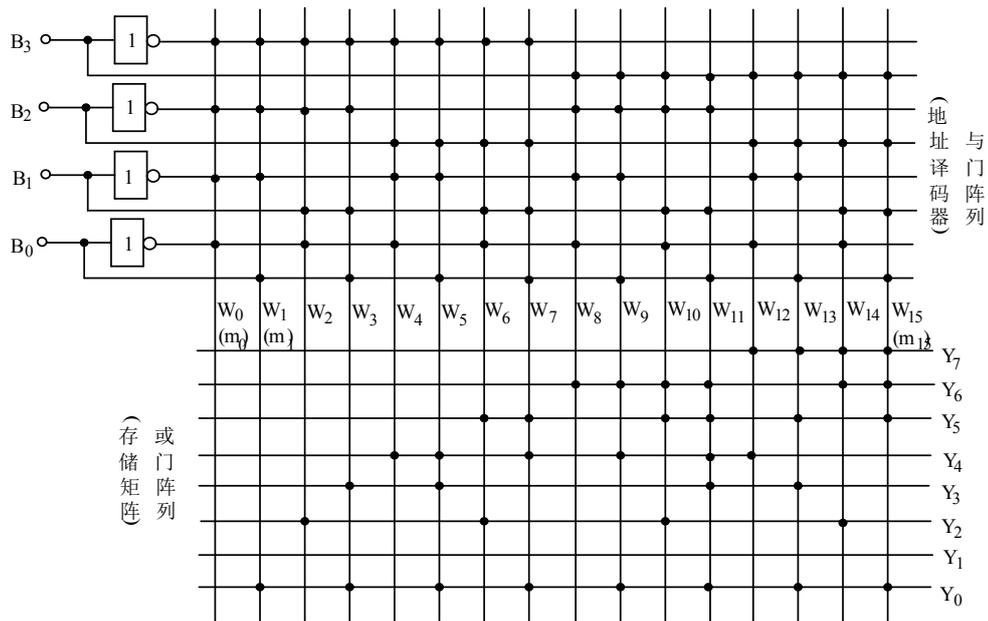


图 1 例 1 中 ROM 存储矩阵连接图

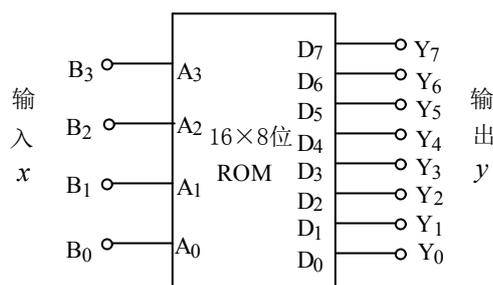


图 2 例 1 中 ROM 的方框图表示方法

## 2. 实现任意组合逻辑函数

从 ROM 的逻辑结构示意图可知，只读存储器的基本部分是与门阵列和或门阵列，与门阵列实现对输入变量的译码，产生变量的全部最小项，或门阵列完成有关最小项的或运算，因此从理论上讲，利用 ROM 可以实现任何组合逻辑函数。

【例 2】试用 ROM 实现下列函数：

$$Y_1 = \overline{ABC} + \overline{ABC} + \overline{ABC} + ABC$$

$$Y_2 = BC + CA$$

$$Y_3 = \overline{ABCD} + \overline{ABCD} + \overline{ABCD} + \overline{ABCD} + \overline{ABCD} + ABCD \quad Y_4 = ABC + ABD + ACD + BCD$$

【解】

①写出各函数的标准与或表达式按  $A$ 、 $B$ 、 $C$ 、 $D$  顺序排列变量，将  $Y_1$ 、 $Y_2$  扩展成为四变量逻辑函数。

$$Y_1 = \sum_m (2,3,4,5,8,9,14,15)$$

$$Y_2 = \sum_m (6,7,10,11,14,15)$$

$$Y_3 = \sum_m (0,3,6,9,12,15)$$

$$Y_4 = \sum_m (7,11,13,14,15)$$

②选用  $16 \times 4$  位 ROM，画存储矩阵连线图（如图 3）

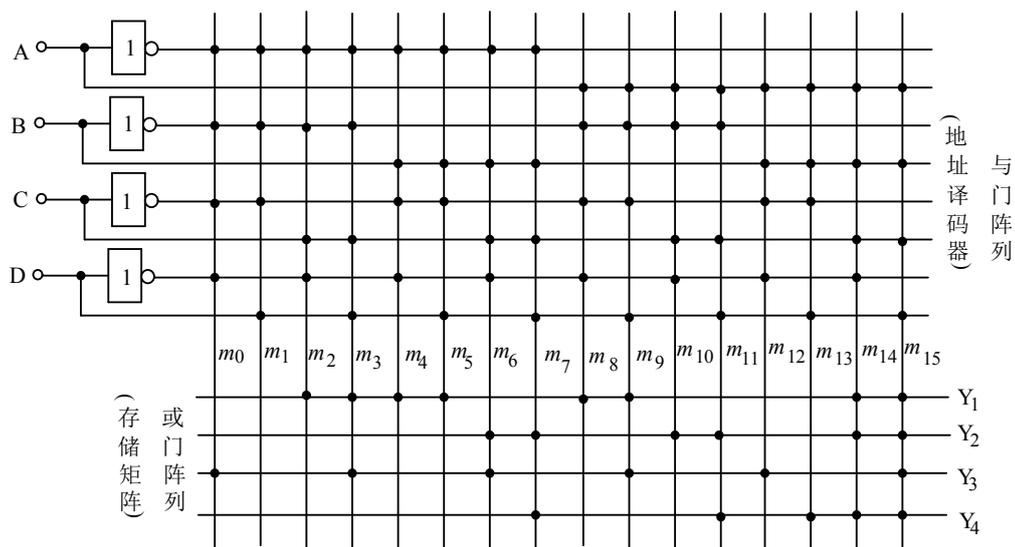


图 3 例 2 中 ROM 存储矩阵连线图